Assignment 1

Practical Data Science (Spring 2020)

Due: 11:59 PM on January 31, 2020.

1 Introduction

In this assignment, you are going to build a small recommendation system. Here, you are given a training set containing ratings of `users` on `items` that they have interacted with. From this, you have to learn a model that predicts the rating, given a user-item pair.

The dataset files are made available at `/home/e0268-master/a1`. Please do not create a local copy for the data files. Please store your code in the directory `/home/e0268-xx/a1`. Your runfile (explained later) must be in this assignment root directory.

2 Specifications

1. The dataset is prepared such that the training set (provided in the file `ratings.train`) and the test set (kept secret) contain roughly 70% and 30% of the total interactions, respectively.
2. A small sample test file containing 10 interactions is provided at `ratings.test` for your reference.
3. Each file begins with a single integer N (the number of interactions in the file), followed by N lines, each containing a single interaction.
4. Use the learned model to predict the ratings for the test set in the same order. The predicted rating should be written out to the specified test file, one per line.
5. Ratings are real values between 0.5 to 5.0, with a step size of 0.5.

2.1 Training input format

Each line after the first contains an interaction combined with a rating, `user`, `item` and `rating` respectively:

```
3
0 7 3.5
1 2 0.5
2 9 5.0
```

3 Evaluation

Your program will be executed by issuing the following command:

```
$ python3 /home/e0268-xx/a1/run.py ⟨test input filename⟩
⟨predicted output fileame⟩
```

Your program must take as input the specified input test file, and produce the output at the path specified in the second argument. Please note that the train file will not be passed at test time.

3.1 Test input filename format

Each line after the first contains an interaction without an associated rating, `user`, `item` respectively:

```
2
0 4
1 9
```

3.2 Predicted output filename format

Write out the `ratings`, one per line (do **not** use scientific notation like `1e-3`):

```
1.75
3.5111111183
```

Do not train your model at test time. Your code will be executed with a time limit of 10 minutes at test time. Please train it ahead of time (there is no time limit for training).

3.3 Assessment

The root mean square error of predictions will be computed:

$$\text{RMSE} = \sqrt{\frac{1}{|\Omega_{\text{test}}|} \sum_{(u,j)\in\Omega_{\text{test}}} (\hat{r}_{uj} - r_{uj})^2}$$

where $\hat{r}_{uj}$ denotes predicted rating of user u on item j, r_{uj} denotes original rating of user u on item j, Ω_{test} is a set denoting the existance of interaction, i.e. $(u, j) \in \Omega_{\text{test}}$ if $r_{uj} \neq 0$.

The marks you obtain will depend on (i) if your code executes successfully, and (ii) if it does, then the performance of your implementation (as against the average performance of the class). **If your code does not run for whatever reason, you will get a 0 in the assignment.**

4 Common Pitfalls

To ensure that your program runs properly, please ensure that you have not made any of the mistakes made by students frequently in past offerings of the course:

- **(Privacy of your code) Don't forget to change your password and read-protect your code from others. You are responsible for safeguarding your code from others.**
- Ensure your filenames are in correct case, in particular `a1/run.py`.
- Make sure you're using Python 3, not Python 2.
- Make sure any additional packages you install are usable without requiring running any activation commands.
- Do not mix `--user` packages along with `virtualenv` packages unless you have prior experience. Stick to one or the other, not both.
- Ensure that your output file is in the correct format.
- Do not hardcode the filenames of the test input and output files, read it from the input arguments.
- Do not hardcode the number of lines in the file, read it from the supplied input file.
- Make sure all required packages are installed.

- **Log in through SSH and verify your code is running, and produces the desired output.** Do not mount the server through SSHFS (e.g. through the File Manager on Ubuntu) and accidentally execute your code locally by using Right click → Open in Terminal. The required packages will not be installed on the server – which will cause execution to fail, and even if the code does run, there may be inconsistencies in results obtained between your local machine (on which you are running code) and the server (where we will be running the code).
- (Advanced) If you wish to use a Python 3.6 install (e.g. through `pyenv`), ensure that it is the global Python set in your userspace and executes when `python3` is invoked. We will not call `python3.6 run.py`.
- (Advanced) You are free to use `conda` and other package managers, but we cannot provide tech support if your code fails to execute properly. Please verify that your code runs immediately after logging in as a user. If you use `pip` and something goes wrong, we will be able to help.

We would recommend that you come back and go through this checklist to ensure you have not committed these mistakes.

5 In closing

If you have any questions or concerns, feel free to ask on Piazza! Please keep all questions to Piazza, and wait for up to 24 hours for a response. Do check if your question has been answered before asking.

Any attempt at plagiarism will result in a 0/8 on this assignment, and a second offense will result in a grade drop. If you make use of any code snippets available online, please leave an inline comment citing the source from which you have taken it. Copying entire repositories is not allowed.